

WHAT IS CLAIMED IS:

1 1. A method of obtaining a scalar value from a vector register for use
2 in a mixed vector and scalar instruction, comprising:
3 providing a vector in a vector register file; and
4 embedding a location identifier of the scalar value within the vector in
5 bits comprising the mixed vector and scalar instruction.

1 2. The method of claim 1, further including defining the mixed scalar
2 and vector instruction such that at least one dedicated position bit is provided
3 in a bit format of the instruction which provides the location identifier of the
4 scalar value within the vector.

1 3. The method of claim 2, wherein a dimension of the vector is 2^n , and
2 further including providing n bits in the bit format for indicating the location
3 of the scalar value within the vector.

1 4. The method of claim 1, further including embedding the location
2 identifier in an op code provided in the instruction.

1 5. The method of claim 4, further including embedding the location
2 identifier in a secondary op code provided in the instruction.

1 6. The method of claim 1, further including using the instruction in a
2 data processor having a paired singles execution unit, wherein two single
3 precision values constitute the vector.

1 7. A data processor, comprising a vector processing unit, a vector
2 register file and an instruction set, wherein the instruction set includes at least
3 one mixed vector and scalar instruction having a bit format in which a
4 location of a scalar value within a vector needed to execute the instruction is
5 embedded.

1 8. The data processor of claim 7, wherein the location of the scalar
2 value is embedded in a secondary op code of the instruction.

1 9. The data processor of claim 7, wherein at least one dedicated bit is
2 provided in the bit format of the instruction to provide the location of the
3 scalar value within the vector.

1 10. The data processor of claim 9, wherein the vector has a dimension
2 of 2^n and n dedicated bits are provided in the instruction to provide the
3 location of the scalar value within the vector.

1 11. The data processor of claim 7, wherein the mixed scalar and vector
2 instruction specifies vector registers for all operands needed to execute the
3 instruction.

1 12. The data processor of claim 1, wherein the vector has a dimension
2 of two.

1 13. The data processor of claim 2, wherein the vector processing unit is
2 a paired singles unit which processes two single-precision floating point
3 values in parallel.

1 14. An instruction for a data processor, comprising a bit format which
2 includes bits designating a first source vector register, bits designating a
3 second source vector register and bits which indicate a location of a scalar
4 operand within a vector register for use in executing the instruction.

1 15. The instruction of claim 14, wherein the bits which indicate the
2 location of the scalar value within the vector register indicate a location
3 within one of the first source vector register and the second source vector
4 register.

1 16. The instruction of claim 14, further including bits which encode a
2 primary op code and bit which encode a secondary op code.

1 17. The instruction of claim 16, wherein the bits which indicate the
2 location of the scalar value within the vector register are embedded in the bits
3 comprising one of the primary op code and secondary op code.

1 18. The instruction of claim 14, wherein the instruction is executable
2 on a microprocessor having a vector processing unit.

1 19. The instruction of claim 18, wherein the instruction is executable
2 on a microprocessor having a vector processing unit in the form of a paired
3 singles unit.

1 20. An information processor, including a decoder for decoding
2 instructions including at least some graphics instructions and at least one
3 paired singles instruction, wherein the decoder is operable to decode a 32-bit

09545182-040700

4 paired singles floating point add instruction, wherein bits 0-5 encode a
5 primary op code of 4, bits 6-10 designate a floating point destination register
6 for storing a pair of 32-bit single-precision floating point values resulting
7 from the paired singles floating point add instruction, bits 11-15 designate a
8 floating point source register storing a pair of 32-bit single-precision floating
9 point values, bits 16-20 designate a further floating point source register
10 storing a pair of 32-bit single-precision floating point values, bits 21-25
11 encode a reserved field of "00000", bits 26-30 encode a secondary op code of
12 21, and bit 31 comprises a record bit indicating updating of a condition
13 register.

1 21. An information processor, including a decoder for decoding
2 instructions including at least some graphics instructions and at least one
3 paired singles instruction, wherein the decoder is operable to decode a 32-bit
4 paired-single-scalar-vector-multiply-add-high (ps_madds0x) instruction
5 wherein a high order word of a paired singles register is used as a scalar, and
6 further wherein the ps_madds0x instruction includes bits 0 through 31,
7 wherein bits 0-5 encode a primary op code of 4, bits 6-10 designate a floating
8 point destination register for storing the results of the instruction, bits 11-15
9 designate a first floating point register as a first source storing a first pair of
10 32-bit single-precision floating point values, bits 16-20 designate a second
11 floating point register as a second source storing a second pair of 32-bit
12 single-precision floating point values, bits 21-25 designate a third floating
13 point register as a third source storing a third pair of 32-bit single-precision

1 24. A decoder for decoding instructions including at least some
2 graphics instructions, wherein the decoder is operable to decode:

3 a 32-bit paired singles floating point add instruction, wherein bits 0-5
4 encode a primary op code of 4, bits 6-10 designate a floating point destination
5 register for storing a pair of 32-bit single-precision floating point values
6 resulting from the paired singles floating point add instruction, bits 11-15
7 designate a floating point source register storing a pair of 32-bit single-
8 precision floating point values, bits 16-20 designate a further floating point
9 source register storing a pair of 32-bit single-precision floating point values,
10 bits 21-25 encode a reserved field of "00000", bits 26-30 encode a secondary
11 op code of 21, and bit 31 comprises a record bit indicating updating of a
12 condition register;

13 a 32-bit paired-single-scalar-vector-multiply-add-high (ps_madds0x)
14 instruction wherein a high order word of a paired singles register is used as a
15 scalar, and further wherein the ps_madds0x instruction includes bits 0 through
16 31, wherein bits 0-5 encode a primary op code of 4, bits 6-10 designate a
17 floating point destination register for storing the results of the instruction, bits
18 11-15 designate a first floating point register as a first source storing a first
19 pair of 32-bit single-precision floating point values, bits 16-20 designate a
20 second floating point register as a second source storing a second pair of 32-
21 bit single-precision floating point values, bits 21-25 designate a third floating
22 point register as a third source storing a third pair of 32-bit single-precision
23 floating point values, bits 26-30 encode a secondary op code of 14 and bit 31
24 comprises a record bit indicating updating of a condition register; and

1 26. A storage medium storing a plurality of instructions including at
2 least some graphics instructions and a 32-bit paired-single-scalar-vector-
3 multiply-add-high (ps_madds0x) instruction wherein a high order word of a
4 paired singles register is used as a scalar, and further wherein the ps_madds0x
5 instruction includes bits 0 through 31, wherein bits 0-5 encode a primary op
6 code of 4, bits 6-10 designate a floating point destination register for storing
7 the results of the instruction, bits 11-15 designate a first floating point register
8 as a first source storing a first pair of 32-bit single-precision floating point
9 values, bits 16-20 designate a second floating point register as a second
10 source storing a second pair of 32-bit single-precision floating point values,
11 bits 21-25 designate a third floating point register as a third source storing a
12 third pair of 32-bit single-precision floating point values, bits 26-30 encode a
13 secondary op code of 14 and bit 31 comprises a record bit indicating updating
14 of a condition register.

1 27. A storage medium storing a plurality of instructions including at
2 least some graphics instructions and a 32-bit paired-single-scalar-vector-
3 multiply-add-low (ps_madds1x) instruction wherein a low order word of a
4 paired singles register is used as a scalar, and further wherein the ps_madds1x
5 instruction includes bits 0 through 31, wherein bits 0-5 encode a primary op
6 code of 4, bits 6-10 designate a floating point destination register for storing
7 the results of the instruction, bits 11-15 designate a first floating point register
8 as a first source storing a first pair of 32-bit single-precision floating point
9 values, bits 16-20 designate a second floating point register as a second
10 source storing a second pair of 32-bit single-precision floating point values,

bits 21-25 designate a third floating point register as a third source storing a third pair of 32-bit single-precision floating point values, bits 26-30 encode a secondary op code of 15 and bit 31 comprises a record bit indicating updating of a condition register.

28. A storage medium storing a plurality of instructions including at least some graphics instructions and:

a 32-bit paired singles floating point add instruction, wherein bits 0-5 encode a primary op code of 4, bits 6-10 designate a floating point destination register for storing a pair of 32-bit single-precision floating point values resulting from the paired singles floating point add instruction, bits 11-15 designate a floating point source register storing a pair of 32-bit single-precision floating point values, bits 16-20 designate a further floating point source register storing a pair of 32-bit single-precision floating point values, bits 21-25 encode a reserved field of "00000", bits 26-30 encode a secondary op code of 21, and bit 31 comprises a record bit indicating updating of a condition register;

a 32-bit paired-single-scalar-vector-multiply-add-high (ps_madds0x) instruction wherein a high order word of a paired singles register is used as a scalar, and further wherein the ps_madds0x instruction includes bits 0 through 31, wherein bits 0-5 encode a primary op code of 4, bits 6-10 designate a floating point destination register for storing the results of the instruction, bits 11-15 designate a first floating point register as a first source storing a first pair of 32-bit single-precision floating point values, bits 16-20 designate a second floating point register as a second source storing a second pair of 32-

21 bit single-precision floating point values, bits 21-25 designate a third floating
 22 point register as a third source storing a third pair of 32-bit single-precision
 23 floating point values, bits 26-30 encode a secondary op code of 14 and bit 31
 24 comprises a record bit indicating updating of a condition register; and
 25 a 32-bit paired-single-scalar-vector-multiply-add-low (ps_madds1x)
 26 instruction wherein a low order word of a paired singles register is used as a
 27 scalar, and further wherein the ps_madds1x instruction includes bits 0 through
 28 31, wherein bits 0-5 encode a primary op code of 4, bits 6-10 designate a
 29 floating point destination register for storing the results of the instruction, bits
 30 11-15 designate a first floating point register as a first source storing a first
 31 pair of 32-bit single-precision floating point values, bits 16-20 designate a
 32 second floating point register as a second source storing a second pair of 32-
 33 bit single-precision floating point values, bits 21-25 designate a third floating
 34 point register as a third source storing a third pair of 32-bit single-precision
 35 floating point values, bits 26-30 encode a secondary op code of 15 and bit 31
 36 comprises a record bit indicating updating of a condition register.